Use of Time-Aware Language Model in Entity Driven Filtering System

Vincent Bouvier
Kware / Aix-Marseille Université
CNRS, LSIS UMR 7296
Domaine universitaire de Saint Jérôme
Avenue Escadrille Normandie Niemen
13397 MARSEILLE Cedex 20
vincent.bouvier@kware.fr

Patrice Bellot
Aix-Marseille Université
CNRS, LSIS UMR 7296
Domaine universitaire de Saint Jérôme
Avenue Escadrille Normandie Niemen
13397 MARSEILLE Cedex 20
patrice.bellot@lsis.org

Abstract

Tracking entities, so that new or important information about that entities are caught, is a real challenge and has many applications (e.g., information monitoring, marketing,...). We are interesting in how to represent an entity profile to fulfill two purposes: 1. entity detection and disambiguation, 2. novelty and importance quantification. We propose an entity profile, which uses two language models. First, the Reference Language Model (RLM), which is mainly used for disambiguation. Second, we propose a formalization of a Time-Aware Language Model, which is used for novelty detection. To rank documents, we propose a semi-supervised classification approach which uses meta-features computed on documents using entity profiles and time series.

1 Introduction

This article introduces the system for the Knowledge Base Acceleration (KBA) track from Text REtrieval Conference (TREC). This challenging task started in 2012 to answer a need in Information Retrieval. Many documents appear everyday on the Web. Finding relevant documents about a topic may be a difficult task depending on the definition of relevancy. The KBA track focuses on filtering documents that are centered on a topic while ranking them according to whether the documents carry important or additional information about the topic.

(Frank et al., 2012) showed that the time lag between the publication date of cited news articles and the date the news is actually written onto the concerned Wikipedia article can be really big (median 356 days) especially for non-popular entities. A possible application is to use highly ranked documents as suggestions for contributor of Wikipedia.

The KBA track is divided in two tasks: CCR (Cumulative Citation Recommendation) and SSF (Streaming Slot Filling). CCR task is to filter out documents worth citing in a profile of an entity (e.g., Wikipedia or freebase article). SSF task is to detect changes on the given slots for each of the target entities. This article focuses only on CCR task.

In CCR task, the system is to filter out, from a stream, the documents relative to target entities. The system must also be able to give the usefulness of a document ranked using one of those 4 relevance classes:
- garbage: no information about target entity;
- neutral: informative but not citable;
- useful: bio, primary or secondary source useful when creating a profile from scratch;
- vital: timely info about the entity's current state, actions, or situation.

The stream-corpus contains timestamped documents crawled from newswires, blogs, forums, reviews,.... The stream corpus must be processed in chronological order in order to perform real life filtering simulation. In addition, the documents relevancy assessment must be performed as soon as the document appears on the stream. A decision cannot be postponed. Each year a set of entities is selected by organizers and a set of documents is annotated according to the selected entities. In 2014, about 30,000 documents have been annotated (8,000 can be used for training purposes).

Our approach uses semi-supervised build entity profile, time series analysis to compute a set of meta-features for each documents. The meta-features are used in a classification system to determine the class of the documents among garbage, neutral, useful and vital. In the remaining of this article we detail the whole concept around entity profile, then we describe the different meta-features used in the classification system. We then detail the different strategies we adopt. We eventually discuss about our experiments onto the KBA framework and the results from the official and unofficial KBA submissions. Unofficial KBA submissions comes from experiments run after the official submission deadline.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2014		**00-00-2014 to 00-00-2014**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Use of Time-Aware Language Model in Entity Driven Filtering System	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Aix-Marseille University ,CNRS, LSIS UMR 7296,Domaine Universitaire St Jerome,13397 MARSEILLE Cedex 20,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
presented in the proceedings of the Twenty-Third Text REtrieval Conference (TREC 2014) held in Gaithersburg, Maryland, November 19-21, 2014. The conference was co-sponsored by the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency (DARPA).

14. ABSTRACT
Tracking entities, so that new or important information about that entities are caught is a real challenge and has many applications (e.g., information monitoring, marketing,...). We are interesting in how to represent an entity profile to fulfill two purposes 1. entity detection and disambiguation 2. novelty and importance quantification. We propose an entity profile, which uses two language models. First, the Reference Language Model (RLM), which is mainly used for disambiguation. Second, we propose a formalization of a Time-Aware Language Model, which is used for novelty detection. To rank documents, we propose a semi-supervised classification approach which uses meta-features computed on documents using entity profiles and time series.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**7**	

2 Entity Profiles

The term *entity* describes a single and unique representation of a person, an organization, music band,.... Documents refer to entities using their surface form names. An entity may have several surface form names (e.g., Tim Cook, the Apple CEO,...). In addition, one surface form may be used for several entities (e.g., Boris Berezovsky the business man or the pianist). Such ambiguous entities are called homonymous. We propose a filtering system based on two steps filtering method for each document: 1. keep the document only if an occurrence of a surface form is found in it; 2. give a class to the document for each entity detected in step 1. We propose an approach that uses entity profiles as well as a classification system to perform those two steps.

2.1 Detecting entities within documents

The first step of the filtering system is aimed to find documents that contain an occurrence of the entity. (Cucerzan, 2007) propose an approach that uses Wikipedia to build an entity profile. Given the entity dedicated Wikipedia Page, the method consists in using heuristics and knowledge base graph exploration to extract: a language model, a list of relations (all entities having a connection to the entity dedicated Wikipedia Page) and a list of surface form names.

The most intuitive way to detect an entity within a document is to find occurrences of any surface form. We propose heuristics to automatically build patterns out of surface form names. Those heuristics are aimed to detect acronyms and middle names. We use the notation `[ ]` to surround optional words. We use $\star$ to announce that the word is incomplete.

```
B.N.S.F. railway   =>  B⋆N⋆S⋆F⋆ railway
Chad R. Kroeger    =>  Chad [R⋆] Kroeger
```

We use this notation in addition to the (Cucerzan, 2007) approach to search for surface forms within the Wikipedia page centered on the entity. For instance, the system can now detect that *B.N.S.F. railway* stands for *Burlington Northern Santa Fe railway*.

2.2 Language Models in Entity Profile

One main aim of the entity profile is to help in entity disambiguation. (Navigli, 2009) shows that having a context, in which a word occurs in, helps in word sense disambiguation. The same observation can be transferred to entities. (Sehgal and Srinivasan, 2007) define an entity profile as a language model build using the top-n documents found on Google. Then, they compare the obtained result with the Wikipedia page corresponding to the entity and obtained good results. However, such method does not address the homonymous issue.

(Efron, 2014) proposed a method to update the language model of an entity profile using documents ranked as relevant by their system. However, results were impacted badly. Indeed, updating the language model, which is used to describe the entity, may lead to a topic drift. To avoid the topic drift, we define an entity profile with two language models serving two different but essential purposes:

- the Reference Language Model (RLM) gathers information aimed to help identifying the entity within documents. The language model is a unigram representation where each word is associated with a probability. To completely avoid the topic drift, a RLM must only be updated with manual inputs. In addition, approaches from (Cucerzan, 2007; Sehgal and Srinivasan, 2007) can be used to easily build such model.

- the Time-Aware Language Model (TALM) catches a representation of current events that occurs for an entity. The language model is a unigram representation where each word is associated with a probability and a timestamp. Contrary to the RLM, the TALM is constantly updated using documents ranked as relevant to an entity. We use the time component and a sigmoid function to forget about information after a certain time laps. We think that two identical events can appear at different time laps and we want to be able to catch both of thus. Indeed the fact that an event is happening again after a period may infer that something important is happening for the entity about that particular event (e.g, someone's getting married several times).

3 Language Models formalization

3.1 The Reference Language Model

The RLM represents the knowledge on the entity. Tthose knowledge helps for entity disambiguation. We propose to use probabilities from the RLM to directly compare them to the document using distance, like the cosine similarity. The distance score indicates whether the context is similar to the one described in the RLM or not.

Let us define $\mathcal{R}$ as a set of documents such as $[d_1, ..., d_n \in \mathcal{R}]$. Let $tf(w_i, d_n)$ the function that gives the number of occurrences of a word w_i in a document d_n. We define $df(w_i, \mathcal{R})$ the number of time a word w_i occurs in the language model $\mathcal{R}$ such as:

$$df(w_i, \mathcal{R}) \quad = \quad \sum_{n=0}^{\mathcal{R}} tf(w_i, d_n) \qquad (1)$$

Let us define the functions $len(d_n)$ the number

of occurrences of each words $[w_1, ..., w_i] \in d_n$ and $len(\mathcal{R})$ the number of occurrences of each words $[w_1, ..., w_i] \in \mathcal{R}$ such as:

$$\begin{array}{l} len(d_n) = \sum_{i=0}^{d_n} tf(w_i, d_n) \\ len(\mathcal{R}) = \sum_{n=0}^{\mathcal{R}} len(d_n) \end{array} \quad (2)$$

The normalized version of term frequency is referred to as the *term probability*. We then define:

$$\begin{array}{lcl} p(w_i|d_n) & = & \frac{tf(w_i,d_n)}{len(d_n)} \\ p(w_i|\mathcal{R}) & = & \frac{df(w_i,\mathcal{R})}{len(\mathcal{R})} \end{array} \quad (3)$$

3.2 The Time-Aware Language Model

The Time-Aware Language Model (TALM) searches for novelty about an entity. The TALM aggregates information from documents being relevant for the entity. However gathering too much information may lead, at a certain point, to miss novelty. In addition, a Language Model with too many information in it may lead to a drift. We design the TALM so that it uses a time-aware function allowing it to smoothly forget old documents. A time-aware function gives a weight according to two events e_1 and e_2 having respectively a timestamp t_{e1} and t_{e2} with $t_{e1} \geq t_{e2}$. We propose $\mathcal{D}$ a time-aware function that gives, to a word, less credit if it was seen a long time ago. The amount of time required to forget about an information is defined using a constant parameter λ as follows:

$$\begin{array}{lcl} \Delta_t & = & \frac{1}{\lambda} * (t_{e1} - t_{e2}) \\ \mathcal{D}(t_{e1}, t_{e1}) & = & \begin{cases} 1, & \text{if } \Delta_t < 0 \\ 0, & \text{if } \Delta_t > 1 \\ \frac{1}{1+e^{(\rho((\Delta_t)-0.5))}} & \text{otherwise} \end{cases} \end{array} \quad (4)$$

Let us define $\mathcal{T}^{\mathcal{A}}$ a time-aware language model made up of a set of timestamped documents such as $[d1 \rightarrow t_{d1}, ..., d_n \rightarrow t_{dn}] \in \mathcal{T}^{\mathcal{A}}$. We use $\_^{\mathcal{A}}$ as an indicator that the function is using in time-aware context. Let us consider d_c a new document having a timestamp t_c. Let $tf^{\mathcal{A}}(w_i, d_n, t_c)$ a function that computes the number of occurrences of words w_i in a document d_n while considering a time t_c. We also define $df^{\mathcal{A}}(wi, t_c, R)$ the number of time a word w_i occurs in $\mathcal{T}^{\mathcal{A}}$ as follows:

$$\begin{array}{lcl} tf^{\mathcal{A}}(w_i, d_n, t_c) & = & \mathcal{D}(t_c, t_{dn}).count(w_i|d_n) \\ df^{\mathcal{A}}(w_i, \mathcal{T}^{\mathcal{A}}, t_c) & = & \sum_{n=0}^{D} tf^{\mathcal{A}}(w_i, d_n, t_c) \end{array} \quad (5)$$

Considering a time t_c, let us define the function $len^{\mathcal{A}}(d_n, t_c)$ the number occurrences of each words $[w_1, ..., w_i] \in d_n$. Let us define $len^{\mathcal{A}}(\mathcal{T}^{\mathcal{A}}, t_c)$ the number of occurrences of each words $[w_1, ..., w_i] \in \mathcal{T}^{\mathcal{A}}$ as follows:

$$\begin{array}{lcl} len^{\mathcal{A}}(d_n, t_c) & = & \sum_{i=0}^{d_n} tf^{\mathcal{A}}(w_i, d_n, t_c) \\ len^{\mathcal{A}}(\mathcal{T}^{\mathcal{A}}, t_c) & = & \sum_{n=0}^{D} len^{\mathcal{A}}(d_n, t_c) \end{array} \quad (6)$$

Let us define $N^{\mathcal{A}}(\mathcal{T}^{\mathcal{A}}, t_c)$ the number of documents considered at time t_c and $idf^{\mathcal{A}}(w_i, t, \mathcal{T}^{\mathcal{A}})$ the inverse documents frequency as follows:

$$\begin{array}{lcl} N^{\mathcal{A}}(\mathcal{T}^{\mathcal{A}}, t_c) & = & \sum_{n=1}^{D} \mathcal{D}(t_c, t_{dn}) \\ idf^{\mathcal{A}}(w_i, t, \mathcal{T}^{\mathcal{A}}) & = & log \frac{N^{\mathcal{A}}(t,\mathcal{T}^{\mathcal{A}})+1}{df^{\mathcal{A}}(w_i,t,\mathcal{T}^{\mathcal{A}})+0.5} \end{array} \quad (7)$$

To define the term probability functions, we need to consider the time t_{wi} corresponding to the last time the word w_i has occurred in $\mathcal{T}^{\mathcal{A}}$. We now define $p^{\mathcal{A}}(w_i, t_{wi}, t_c|d_n)$ and $p(^{\mathcal{A}}w_i, t_{wi}, t_c|\mathcal{T}^{\mathcal{A}})$ the term probability functions as follows:

$$\begin{array}{lcl} p^{\mathcal{A}}(w_i, t_{wi}, t_c|d_n) & = & \mathcal{D}(t_c, t_{wi}).\frac{tf'(w_i,d_n)}{size(d_n)} \\ p^{\mathcal{A}}(w_i, t_{wi}, t_c|\mathcal{T}^{\mathcal{A}}) & = & \mathcal{D}(t_c, t_{wi}).\frac{\sum_{n=0}^{D} p(w_i,t_{wi},t_c|d_n)}{\sum_{n=0}^{D} \mathcal{D}(t_c,t_{dn})} \end{array} \quad (8)$$

4 Documents classification using meta-features

In the previous year of KBA, many systems have been using meta-features within a classification system (Bonnefoy et al., 2013a; Bonnefoy et al., 2013b; Balog et al., 2013; Bouvier and Bellot, 2014). Those study show that some meta-features works better than others. We summarize in the following subsections the meta-features we have been using as well as the new features designed with our new entity profile representation.

4.1 Entity Disambiguation meta-features

The entity related meta-features are aimed to quantify, using different measures, how a document is relevant to the entity. In the first filtering step, a document is selected using only the surface form names. However, an entity can be ambiguous and thus a document may contains occurrences of surface form names of an homonymous entities.

To ensure a document refers to the target entity, we use the context given by the entity profile to compute the following features:

- The Cosine Similarity is computed using the term frequency $tf(w_i|V)$ of words $w_i \in d \bigcup \mathcal{R}$ given the vector representation of the document d and the Reference Language Model (equation 9);

$$cos(d, \mathcal{R}) = \frac{\sum_{i=1}^{n} tf(w_i|d).tf(w_i|\mathcal{R})}{\sqrt{\sum_{i=1}^{n} tf(w_i|d)^2}.\sqrt{\sum_{i=1}^{n} tf(w_i|\mathcal{R})^2}} \quad (9)$$

- The Surface Forms Term Frequency measures the term frequency of each surface forms within the document and the title;
- Entity Relations Term Frequency measures the term frequency of each relations by type of relations (incoming, outgoing, mutual) extracted from the knowledge graph from Wikipedia.

4.2 Novelty and Importance meta-features

We propose to use the Time-Aware Language Model (TALM) we formalize in section 3.2 to catch novelty by using different known measures :
- Jensen Shannon Divergence (JSD) computes divergence between two vectors. It uses a third vector $\mathcal{M}$ resulting from averaging the dot products of the two vectors to compare. Let us define a set of n words $w_i \in d \bigcup \mathcal{T}^{\mathcal{A}}$ considering TALM $\mathcal{T}^{\mathcal{A}}$ and a document d appearing at time t_d, JSD can be written such as:

$$\begin{aligned} \mathcal{M} &= \tfrac{1}{2} * (d + \mathcal{T}^{\mathcal{A}}) \\ JSD &= \tfrac{1}{2} * \sum_i^n p^{\mathcal{A}}(w_i, t_d|\mathcal{T}^{\mathcal{A}}) log \frac{p^{\mathcal{A}}(w_i, t_d|\mathcal{T}^{\mathcal{A}})}{p(w_i|\mathcal{M})} \\ &\quad + \tfrac{1}{2} * \sum_i^n p(w_i|d) log \frac{p(w_i|d)}{p(w_i|\mathcal{M})} \end{aligned} \quad (10)$$

- Time-Aware Novelty Score given by (Karkali et al., 2014). They have tested different approaches to measure novelty on real world dataset. The novelty score that outperforms others is computed using a smoothed version of the well known $tf.idf$ weighting scheme with time components. We transcribe the equation so that we can use it with the TALM (equation 11).

Burst detection have been used in event detections or forcasting (Kleinberg, 2002; Sakaki et al., 2010; Weng and Lee, 2011). It has been shown in (Amodeo et al., 2011; Peetz et al., 2014; Wang et al., 2007) that the relevancy of search results can be improved using timed information such as abnormal peaks (bursts) of queries in log files or of keywords or even documents related to an entity in a stream. There are diverse reasons to explain a burst. Figure 1 shows a burst when an important event occurs concerning the entity BNSF Railway. The meta-features we propose to use for importance quantification are:
- The Kleinberg Burst measures;
- The Elastic Burst measures that uses wavelet trees to estimate burst strength;

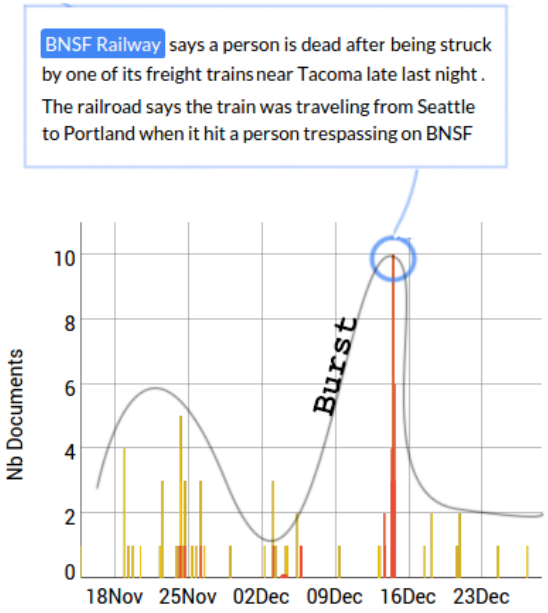

Figure 1: Showing a burst of documents corresponding to an important news about BNSF Railway.

5 Experimental Setup

The Entity profiles are build as a pre-process step using a dump of Wikipedia from january 2012 in addition to the reference files provided for each entities. Thus, each entity have:
- a Reference Language Model (RLM) initialized. The RLM can be empty if no reference file has been provided;
- a set of relations (incoming, outgoing, mutual) found using Wikipedia knowledge graph exploration. In the case where no Wikipedia page were found for an entity, the set remains empty;
- a set of surface forms found using heuristics from (Cucerzan, 2007) and the pattern recognition introduced in section 2.1;
- an empty Time-Aware Language Model, which is filled while going through the stream-corpus.

Finally, each documents $[d_1, ..., d_n] \in S$ from the stream S is processed according to the two filtering steps:

1. The document d_n contains an occurrence of a surface form from one or several entities. The document is evaluated for each entity detected in it. Otherwise, the document is not evaluated;

2. For each entity detected in the document d_n, the meta-features are computed and the classification system output the relevancy of the document as well as a confidence score. The rel-

$$NS^{\mathcal{A}}(d, t_d, \mathcal{T}^{\mathcal{A}}) = \frac{1}{\sum_{i=0}^{d} tf^{\mathcal{A}}(w_i, t, \mathcal{T}^{\mathcal{A}})} * \sum_{i=0}^{d} tf^{\mathcal{A}}(w_i, t_d, \mathcal{T}^{\mathcal{A}}).idf^{\mathcal{A}}(w_i, t_d, \mathcal{T}^{\mathcal{A}}) \quad (11)$$

evancy and the score is stored in the final run submissions.

We define different strategies to compute meta-features. Indeed, each entity profile is made up of a TALM that has to be updated with documents. Documents may contain noise that we don't want to be reflected in the TALM. We use two different strategies to update the TALM:
- Update with Document (UD): the TALM is updated with the full document;
- Update with Snippet (US): the TALM is updated only with the paragraph that contains occurrences of the entity;
- No Update (NU): the TALM (and the meta-features associated to it) are not used in order to see if it brings any value to our system.

For the classification system, we use a Random Forrest classifier with 50 trees. We designed four different classification strategies:
- the first strategy, *2STEPS*, considers the problem as a binary classification problem where we use two classifiers in cascade. The first one $C_{GN/UV}$ is to classify among two classes: Garbage/Neutral and Useful/Vital. For documents being classified as Useful/Vital a second classifier $C_{U/V}$ is used to determine the final output class between Useful and Vital; - the second strategy, *SINGLE*, performs directly a classification between the four classes; - the third strategy, *VvsAll*, trains a classifier on all documents considering only two classes vital and others (all classes but vital). When this classifier gives a non-vital class, the SINGLE method is used to determine another class among Garbage, Neutral and Useful; - the last strategy, *MULTI*, uses scores emitted by all previous classifiers and learns the best output class considering all classifier's scores for every classes.

We submit 9 runs where each run explore a combination of update strategy combine with a classification strategy.

6 Results analysis

We propose a two step filtering system where the first step is aimed to keep only documents having an occurrence of a surface form of at least one entity. To measure the performance of the first step we draw the table 1. Those results show that we obtain satisfactory performance since 87% of documents concerning the entities are found with only about 6% of error rate.

Found and in truth data:	87.10%
Found and not in truth data:	5.90%
Not found:	12.9%

Table 1: First filtering step results

Systems	**F-measure Vital**		
	NU	**UD**	**US**
MULTI	.321	**.326**	.316
SINGLE	.252	.261	**.290**
2STEPS	.248	**.304**	.292
VvsAll	.217	.224	**.297**
	F-measure Vital+Useful		
MULTI	.777	**.783**	**.783**
SINGLE	.764	.779	**.784**
2STEPS	.759	.771	**.782**
VvsAll	**.690**	.692	**.720**

Table 2: Scores obtained on our systems MULTI, SINGLE, 2STEPS and VvsOthers with different settings NO-UPD, UPD-DOC, UPD-SNPT for vital and vital+useful classification

The second filtering step consists in giving a class to every documents kept from the first step. For the official submission, we designed 9 different strategies and we obtain the results summarized in table 2. The official measure is the f-measure (harmonic mean of precision and recall). We clearly see that getting satisfactory results for vital document is really difficult. However we obtain almost .80 of f-measure on filtering useful and vital documents. This means that our system is able to depict that a document is centered on an entity at a rate of 80%.

After the submission deadline we found a bug in our first filtering step. Some patterns were not working properly then some documents were missing. After running the system again, the first step performances have then increased. We obtained similar f-measure results for the step 2.

Found & in truth data	98.95%	+11.85%
Found & not in truth data	8.89%	+2.99%
Not found	1.05%	-11.85%

Table 3: First filtering step results with bug fixed on surface forms patterns

For many entities we have just a few information. We wanted to measure if performance could be in-

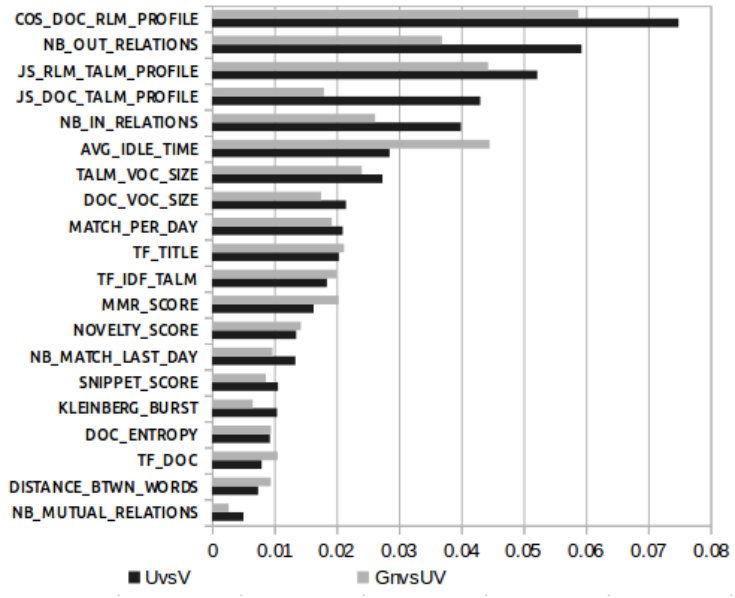

Figure 2: Showing a burst of documents corresponding to an important news about BNSF Railway.

crease with some more knowledge for all entities. We set a limit of 5 reference documents for each entities. Some already have reference documents given by KBA organizers. We add up to 5 useful documents from the training to each entity. We use the first 5 documents seen for each entity. By doing so, we upgrade the profile with more knowledge while still having a scalable system. We obtained the results summarized in table 4. As we can see performances have been widely increased for both useful and vital filtering.

Systems	**F-measure Vital**		
	NU	**UD**	**US**
MULTI	**.387**	.381	.364
SINGLE	**.346**	.337	.307
2STEPS	**.351**	.301	.315
VvsAll	**.339**	.327	.301
	F-measure Vital+Useful		
MULTI	.894	**.895**	.891
SINGLE	**.902**	.892	.893
2STEPS	.890	.889	**.894**
VvsAll	**.895**	**.895**	.891

Table 4: Scores obtained on our systems MULTI, SINGLE, 2STEPS and VvsOthers with different settings NO-UPD, UPD-DOC, UPD-SNPT for vital and vital+useful classification

In order to observe the impact of each features on the classification, we look at the Variable Importance (VI) given by (Breiman, 2001). The VI indicate how significant is a feature in classification decision by randomly changing the values associated to each feature (one at a time) and observing the out of bag error. We show from figure 2 that the Reference Language Model (RLM) and the Time-Aware Language Model (TALM) are among the top 5 important features. In addition, relations discovered on the Wikipedia page of the entity (OUT_RELATIONS) are also very decisive. Finding known relation within a document helps discovering Vital information. On the negative side, we noticed that burst detection does not really helps in finding Vital information which is counter intuitive.

7 Conclusion and Perspectives

To conclude, we present a filtering system based on two filtering steps. We demonstrate that the first step obtained very good results in documents preselection. We see that we obtain satisfactory results based on current KBA-Framework. We also show that the results could be widely increased when having more knowledge about an entity while still having a scalable system. Finally we discovered that the meta-features linked to the Reference Language Model and the Time-Aware Language Model were really useful in vital document classification.

We noticed that burst detection is not always a reliable clue depending on the entity. In the future, we will invest on whether some features correspond more to some entities than others to automatically choose the appropriate system.

References

Giuseppe Amodeo, Giambattista Amati, and Giorgio Gambosi. 2011. On relevance, time and query expansion. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 1973–1976.

Krisztian Balog, Heri Ramampiaro, Naimdjon Takhirov, and Kjetil Nørvåg. 2013. Multi-step classification approaches to cumulative citation recommendation. In *Open research Areas in Information Retrieval, OAIR '13, Lisbon, Portugal, May 15-17, 2013*, pages 121–128.

Ludovic Bonnefoy, Vincent Bouvier, and Patrice Bellot. 2013a. Lsis/lia at trec 2012 knowledge base acceleration. In *The Twenty-First Text REtrieval Conference (TREC 2012) Proceedings*, pages SP 500–298.

Ludovic Bonnefoy, Vincent Bouvier, and Patrice Bellot. 2013b. A weakly-supervised detection of entity central documents in a stream. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*, pages 769–772. ACM.

Vincent Bouvier and Patrice Bellot. 2014. Filtering Entity Centric Documents using Profile Update and Random Forest Classification. In *The Twenty-Second Text REtrieval Conference (TREC 2013) Proceedings*, pages SP 500–302.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 708–716.

Miles Efron. 2014. The university of illinois' graduate school of library and information science at trec 2013. In *The Twenty-Second Text REtrieval Conference (TREC 2013) Proceedings*, pages SP 500–302.

John R Frank, Max Kleiman-Weiner, Daniel A Roberts, Feng Niu, Ce Zhang, Christopher Ré, and Ian Soboroff. 2012. Building an entity-centric stream filtering test collection for trec 2012. In *The Twenty-First Text REtrieval Conference (TREC 2012) Proceedings*, pages SP 500–298.

Margarita Karkali, François Rousseau, Alexandros Ntoulas, and Michalis Vazirgiannis. 2014. Using temporal idf for efficient novelty detection in text streams. *CoRR*, abs/1401.1456.

Jon M. Kleinberg. 2002. Bursty and hierarchical structure in streams. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pages 91–101.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2).

Maria-Hendrike Peetz, Edgar Meij, and Maarten de Rijke. 2014. Using temporal bursts for query modeling. *Inf. Retr.*, 17(1):74–108.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 851–860, New York, NY, USA. ACM.

Aditya Kumar Sehgal and Padmini Srinivasan. 2007. Profiling topics on the web. In *Proceedings of the WWW2007 Workshop I^3: Identity, Identifiers, Identification, Entity-Centric Approaches to Information and Knowledge Management on the Web, Banff, Canada, May 8, 2007*.

Xuanhui Wang, ChengXiang Zhai, Xiao Hu, and Richard Sproat. 2007. Mining correlated bursty topic patterns from coordinated text streams. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007*, pages 784–793.

Jianshu Weng and Bu-Sung Lee. 2011. Event detection in twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*.